Appendix G

Title:
Macro Specification
TB Version 2.1

1 Introduction

1.1 Overview

The Transmit Buffer (TB) provides channelwise buffering of raw data/status words between a Data Management Unit Transmit (DMUT) and a Protocol Machine Transmit (PMT). This data is stored in form of (internal) linked lists for all logical channels. These linked lists are pre-allocated according to bandwidth requirements of the respective channels. In order to avoid transmit underrun conditions each channel buffer has two control parameters for smoothing the filling/emptying process (Transmit Threshold, Request Threshold)

The transmit enable threshold value has following impact:

- Transmission of channel data is started if more than the transmit threshold number of words are he stored in the internal transmit buffer or if at least one word with a complete indication (CI bit, see chapter ...) is stored in channel buffer. Otherwise an empty indication (TB_PTEMPTY) will be delivered to PMT. This feature is always active. Typically after a configuration command or after 'TB empty' condition, but also while frame oriented data transfer this feature reduces the probability of data underrun.

The burst threshold value has following impact:

- As soon as the amount of empty channel buffer locations gets above the threshold value the TB will request data from the transmit DMU (DMUT) .

TB works mode and frame independent. TB transfers data (even the status bit) absolutely transparent and fulfills a channelwise buffering and an efficient data burst request generating.

1.2 Features incl. performance, number of gates

- Number of gates: depends strongly on TB dimensions
- Performance: The overall limitation of data throughput is the PMT interface: each 5th cyle a read access to TB is allowed.
- SMIF register interface or alternatively Target FPI interface for configuration of TB (transmit init and transmit off commands), system test (read of programmable parameter 'burst threshold', 'transmit threshold' and 'channel buffer size' via channel debug command) and testmode (read/write of all internal rams via indirect test register access)
- Interface to DMUT and PMT is FPI like
- interrupt controller (IC) interface with channelwise interrupt generation (configuration fail interrupt, not maskable)
- Programmable FIFO size for each channel (ITBS)
- Programmable burst threshold for DMUT request generation (BTC) and transmit enable threshold (TTC)

1.3 System IntegrationSystem Integration and Application

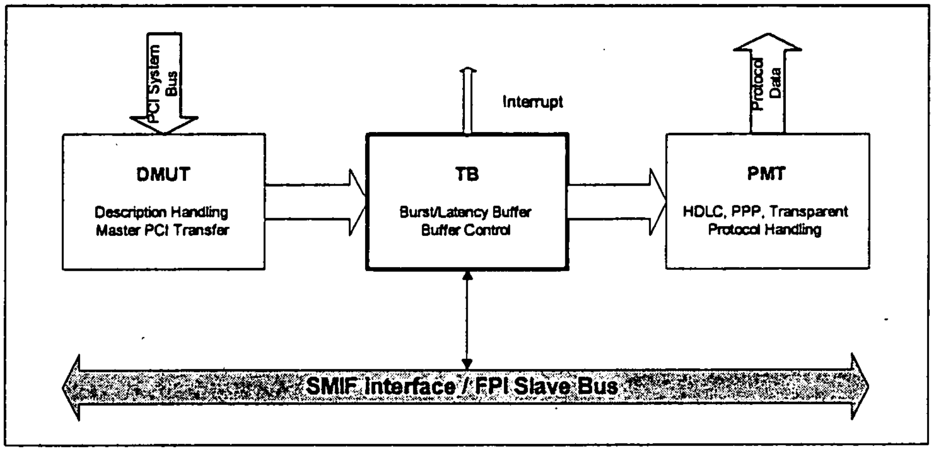


Figure 2
System Integration

The TB has four interfaces:

- SMIF / FPI Slave Interface for programming and system test
- DMUT interface for burst capable data transfer from external memory
- PMT interface for protocoll interface
- DMUI interrupt interface

1.4 Known Restrictions and Problems

The maximum data throughput depends on data bus width, system clock frequency and the programmed configuration of TB.

PMT limitation: The maximum speed of PMT interface is 1 data transfer per 5 cycles. This is a short term maximum througput. TB is able to handle an average througput of 1 data per 20 clock cycles.

With higher burst thresholds and less number of activated channels this average increases.

A further limitation is the behaviour of TB-DMUT-interface: DMUT allows no interleaving of burst request and data transfer, i.e. first a read of request register is done, then either the data are served or the request is stored in DMUT (e.g. HOLD condition in DMUT, see DMUT specification). Afterwards the read of request register is done.

2 Functional and Test Description

2.1 Block Diagram incl. Clocking Regions

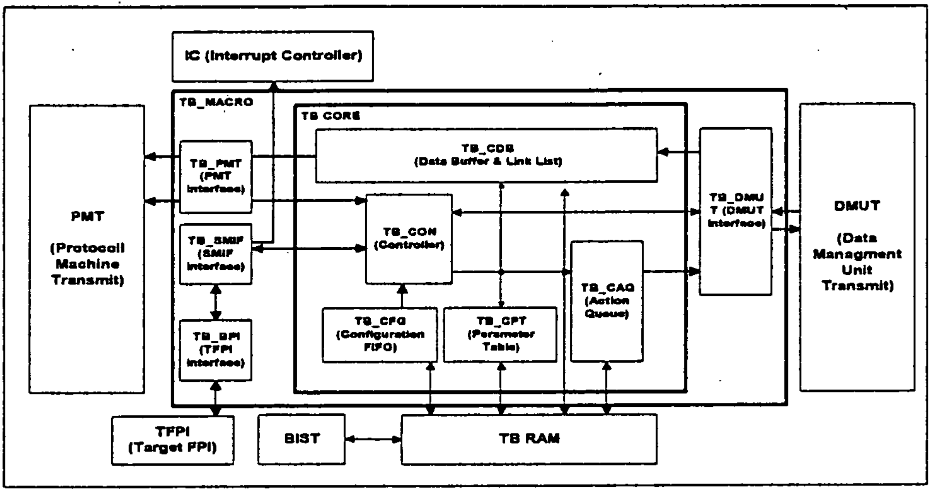


Figure 3
TB Block Diagram

As described in the **Figure 3**, the Transmit Buffer consists of

- Central Data Buffer and link list (TB_CDB)
- Central Parameter Table (TB_CPT) and configuration FIFO (TB_CFG)
- Central Action Queue (TB_CAQ)
- Controller (TB_CON).

Those blocks build the core functionality. Aditionally TB has a FPI like DMUT (TB_DMUT) and a FPI like PMT (TB_PMT) interface. TB macro has an internal SMIF interface (TB_SMIF). Therefore TB offers both

- SMIF and
- TFPI interface (TB_BPI implementation is an optional part of TB Macro).

2.2 Normal Operation Description

After reset all rams are initialized: each ram address is written with idle values and a global link list is set. If initialisation is ready, TB asserts TB_IIP inactive.

Afterwards TB could be configured via TFPI/SMIF interface: a channel command consists of 2 write accesses (TB point of view): first the channel specification buffer register (CSPEC_BUFFER) has to be written with the buffer parameters, then a write of the channel specification command register (CSPEC_CMD) specifies the channel number and the desired command (init, off).

All commands are written into a configuration fifo. This fifo can't be read before GC_STOP is deasserted. Therfore TB won't generate a data request and can't transfer data to PMT during this 'stop' phase.

When GC_STOP is inactiv, the configuration fifo is read.

A transmit init command leads to an allocation of the requested buffer size (ITBS) and writes the according channel number into a action queue. When this action queue is read a data request to DMUT is asserted. After an init command, TB always requests a number of ITBS data from DMUT. Later, the number of requested data will be a function of BTC.

A transmit off command deactivates the channel and deletes all stored channel data. The previous allocated buffer locations become available. The next request to DMUT indicates with DEL=1, that this channel is switched off and all stored requests in DMUT have to be cancelled. RTL has to be ignored and no further requests are generated.

As long as no or less then TTC data are stored in TB, each PMT request is answered with an TB_EMPTY indication.

Incoming data from DMUT will be stored in TB data buffer (TB_CDB) by using the first element from a free pool list. A read request from the PMUT causes the TB to read next data in the channel data buffer (link list) and to add the read location to the free pool list.

Table 1
Request Register for data request from TB to DMUT

Bit	P1			P2	P3			P4	P5
Function	DEL	Reserved		RTL (requested transfer length)		Reserved		CHN (Channel number)	

Table 2
Request Register for data transfer acknowledge from DMUT to TB

P1: tb_dt_crdel_pb_c
P2: tb_dt_crbl_lb_c
P3:tb_dt_crbl_rb_c

Bit	P6			P2	P3			P4	P5
Function	CI	Reserved		STL (served transfer length)		Reserved		CHN (Channel number)	

P4:tb_dt_crchn_lb_c
P5:tb_dt_crchn_rb_c
P6:tb_dt_crci_pb_c

The status of each logical channel is stored in the Parameter Table TB_CPT. As soon as the amount of data stored in the TB gets below the programmable threshold (buffer size minus burst size) for a specific channel, an entry to the action queue (TB_CAQ) is made. TB_CAQ is a FIFO which is dimensioned according to the number of channels (depth = number fo channels). For each channel only one entry is possible. Only the channel number is stored. When the next entry of TB_CAQ is read, a request TB_DTREQ from TB to DMUT is genearated with the number of actual empty buffer locations. Both channel number (CHN) and the number of words (RTL) are written to the request register (read access, refer to **Table 1**). After TB_DTREQ the DMUT has to read the request register and then read the corresponding number of data via external bus interface (PCI) and write them back to TB: before data transfer DMUT writes channel number (CHN) and the number of served length (STL) into request register (write access, refer to **Table 2**). If the last word of the data transfer will be a word that allows data transfer to PMT independent of number of data stored in TB, CI is activated. In some application a frame end (FE) or abort (TAB) indication could force to set CI. CI=1 enables transmission of all data stored in TB.

Unserved or uncompletely served TB requests are stored in DMUT. In HOLD condition TB will continue to request data from DMUT if the request condition is valid. The unserved requests are stored in DMUT. After transfer of the remaining data to PMT TB will become empty. Afterwards TB sets the TB_EMPTY line to PMT. PMT will decide if it is an abort condition or not (in the first case generate an underrun interrupt and in the second case send the idle character). This channel remains inactive as long as the host CPU does not request an activation of DMUT. After activation of DMUT all stored requests have to be served to TB. In case of no activation a 'transmit off' switches this channel to 'idle' condition and gives all reserved buffer locations free.

Status bits and flags have no impact to TB behaviour. E.g. FE (frame end), TAB (transmit abort) and BE (byte enabales) bits are transferred transparently from DMUT through TB to PMT as all other data.

Transfers to the PMT are performed with single words for each channel. The TB provides a control flag (TB_PTSTAT) indicating the word type.

2.2.1 TB Parameter Table (TB_CPT)

The TB_CPT RAM provides a control word per channel for buffer management:

- Channel burst and transmit enable threshold, buffer size
- Count of empty words in channel's buffer chain not already sent to DMUT
- Count of stored words in channel's buffer chain
- Pointers to start and end of buffer chain (CDB read and write addresses)
- Complete channel buffer status and complete indication pointer

RAM size:

MaxNumChan (MNC) words X (6N + BTB + TTB + 4) bit RAM

Table 3: All channel parameters:

ITBS	BTC	TTC	FIRST	AQE	WAIT	CIPV	CIP	FCTR	ECTR	WP	RP
(N)	(BTB)	(TTB)	(1)	(1)	(1)	(1)	(N)	(N)	(N)	(N)	(N)

ITBS: individual transmit buffer size

total buffer size reserved for channel. Max size is $(2^{(N)}-1)$ entries, min. size is 1 entry. Summary of all buffers is DBS entries.

BTC/TTC: burst threshold code/tranmsit enable code

FIRST: wait for first data after configuration

The first data entry after configuration has special conditions

AQE: action queue entry

action queue entry (only one entry per channel allowed);

WAIT: transmission wait until transmit enabled

Channel start of transmit disabled until threshold reached or one complete frame is stored in channel buffer.

CIPV: pointer to last data with a complete indication flag (CI) is valid.

CIP: pointer to last data with a complete indication flag (CI).

FCTR: full counter

number of filled locations in channel buffer. If FCTR = 0 an TB_EMPTY indication is set.

ECTR: empty counter

number of empty locations in channel buffer. If ECTR is bigger than threshold, an action queue entry has to be made

RP: read pointer

Pointer for PMT data requests; RP shows next location to be read.

WP: write pointer

Pointer for DMUT data transfers; WP shows to last written location.

TB_CPT block also includes a free pool pointer (**FPP**) and a free pool counter (**FPC**)..

note: N is dependent on number of words DBS in the CDB RAM.
$2^{N-1} < DBS <= 2^{N}$

2.2.2 TB Data Buffer and link list (TB_CDB)

The TB_CDB size (DBS) is influenced by:

- number of channels supported
- sum of channel bit rates
- thresholds
- maximum bus latency

The data buffer locations are connected to a logical buffer by a pointer link list. The link list has the width N and the depth DBS. Each entry specifies the next data buffer element.

2.2.3 TB Action Queue (TB_CAQ)

FIFO to buffer requests from TB to the DMU Transmit. An entry specifies only the channel number:

Size: number of channels x channel bus width

2.2.4 TB Configuration (TB_CFG)

FIFO to buffer the configuration requests from TFPI/SMIF,

Table 4: All locations:

ITBS (N)	BTC (BTB)	TTC (TTB)	CHN (CHN)

A buffer create command forces an entry with ITBS>0, a buffer delete command forces an entry with ITBS=0.

Size: number of channels x (N + BTB + TTB + CHN)

ITBS: individual transmit buffer size

total buffer size reserved for channel. Max size is $(2^{(N)}-1)$ entries, min. size is 1 entry. Summary of all buffers is DBS entries.

BTC/TTC: burst threshold code/transmit enable code

CHN: channel number

2.3 Reset Behavior

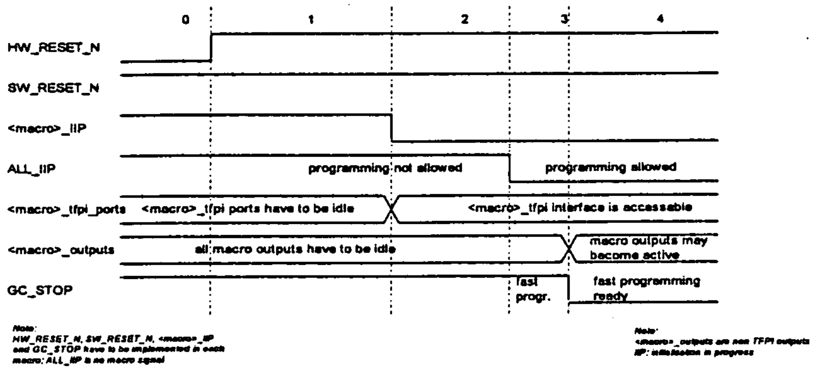
The TB is reset with the line RESET_N.

All internal registers are reset while reset condition; after reset all rams are initialized to known states (RESET state), while TB_IIP is active. The initialisation time depends on the complete data buffer size, because an initial link list in the link list ram has be to generated.

After initialisation TB_IIP becomes inactiv and transmit buffer TB is ready for configuration and data transfer.

After reset/initialisation all interfaces (PMT, DMUT, IC) are inactive and the registers are accessible via the TFPI bus / SMIF for configuration. For a fast configuration the line GCSTOP can be activated to suppress all other interface communication.

Initialisation of Toplevel (macro view)


0
1
2
3
4
HW_RESET_N
SW_RESET_N
<macro>_IIP
ALL_IIP
programming not allowed
programming allowed
<macro>_tfpi_ports
<macro>_tfpi ports have to be idle
<macro>_tfpi interface is accessable
<macro>_outputs
all macro outputs have to be idle
macro outputs may become active
GC_STOP
fast progr.
fast programming ready
Note:
HW_RESET_N, SW_RESET_N, <macro>_IIP and GC_STOP have to be implemented in each macro; ALL_IIP is no macro signal
Note:
<macro>_outputs are non TFPI outputs
IIP: initialisation in progress

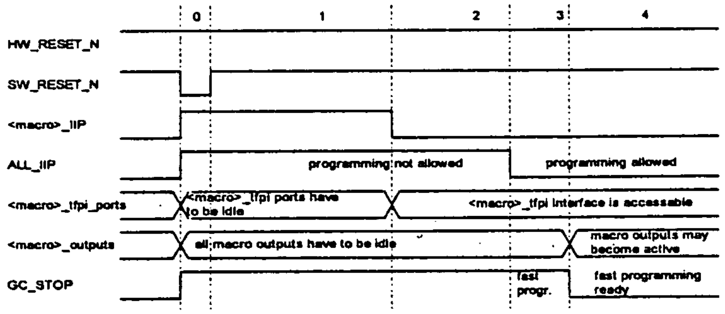
0: HW_RESET_N is active and deactivates SW_RESET_N; all macros are reset; GC_STOP set to '1'

1: HW_RESET_N inactive => each macro executes the macro specific initialisation of rams

2: some macros are ready with initialisation of rams, some not (ALL_IIP is a or-function of all <macro>_IIPs). The "ready macros" could be accessed by TFPI interface, but software programming is not allowed as long as ALL_IIP is active

3: all macros are ready with initialisation => ALL_IIP becomes inactive; software polls the ALL_IIP bit after HW_RESET_N; if ALL_IIP = 0 then software programms all macros (in a default "fast programming mode", GC_STOP = 1; reset value of GC_STOP is '1', but can be reset at any time)

4: after fast programming macro outputs (non TFPI) may become active and macro has to react on non FPI inputs



0: SW_RESET_N becomes active; all macros (registers) are reset; GC_STOP is active

1: SW_RESET_N inactive => each macro executes the macro specific initialisation of rams

2: some macros are ready with initialisation of rams, some not (ALL_IIP is a or-function of all <macro>_IIPs). The "ready macros" could be accessed by TFPI interface, but software programming is not allowed as long as ALL_IIP is active

3: all macros are ready with initialisation => ALL_IIP becomes inactive; software polls the ALL_IIP bit after SW_RESET_N; if ALL_IIP = 0 then software programms all macros in a "fast programming mode", GC_STOP = 1

4: after fast programming macro outputs (non TFPI) may become active and each macro has to react on non FPI inputs

Figure 4
Reset Concept

3 Interfaces and Signal Description

The target of the TB macro are devices that use FPI bus for internal interfaces.

All signals are active high until otherwise specified. Active low signals are designated by "_N" (FPI mode) appended to their names. To make the design as re-usable as possible, a bus signal whose width is application dependent is specified with one of the following parameters:

Parameter name	Bus Type
CNB	Channel Number Bus
DBB	Data/Status Bus
TTC	Threshold Transmit Code
BTC	Threshold Burst Code
ITBS	Buffer Size Bus
RTL/STL	Request/Served Burst Length Bus
N	Link List Pointer Bus/ Data Buffer Address Bus
TFPIAB	TFPI address bus (MSB)

3.1 Signal Description

In the following sections, "Flexible Peripheral Interconnect (FPI) Bus compliant" means that the specified bus uses a subset of the FPI features and satisfies the basic address and data cycle. Not all FPI signals are implemented because default values are sufficient for the application i.e. they can be coded as constants in the hardware. Refer to the FPI bus specification for details of the complete bus.

The following tables lists the FPI-Bus signals and two additional out-band signals:

- TB_STAT to indicate to PMT if transferred word is status instead of pure data. Captured by PMT during data phase.
- TB_EMPTY to indicate to PMT when the TB has no data stored for requested channel.
- TB_REQ_N to indicate that at least for one channel the number of empty cells has reached the programmed burst threshold size.
- DTSTAT while data phase of DMUT to indicate if transferred word is status instead of pure data. Status bit is transferred transparently through TB to PMT.

Table 5
Macro Interfaces and Signal Description

Symbol name	I/O	Function
Clock and Reset		
SYSCLK	I	Internal system clock (66 MHz)
RESET_N	I	General reset of TB. All registers and RAM reset or initialization.
GCSTOP	I	Stop all non configuration process in TB (fast programm mode)
TB_IIP	O	Initialization of TB rams in progress
Protocol Machine Receive (PMT) Interface		
PTRD_N	I	PMT read. Only single word write transfer is supported.
PTA [CNB-1:0]	I	Address bus. Specifies channel number for transfer.
TB_PTD [DBB-1:0]	O	TB Data/Status word being transferred to PMT.
TB_PTRDY	O	Ready. End of data transfer indication. 0 => TB inserts wait state. 1 => TB will finish transfer during next clock cycle.
TB_PTSTAT	O	additional status bit
TB_PTEMPTY	O	Asserted for one cycle while PMT read if TB has no data stored for the channel specified by the address bus.
DMU Transmit (DMUT) Interface		
DTA	I	Address bus (1 bit) 0 => Request register. 1 => Data register.
DTRD_N	I	DMUT Read (of request register)
DTWR_N	I	DMUT Write (of request register or data register)
DTD[DBB-1:0]	I	Input for request register of data register

Table 5
Macro Interfaces and Signal Description (cont'd)

Symbol name	I/O	Function
DTSTAT	I	Status indication from DMUT
TB_DTREQ_N	O	Service request from TB to DMUT controller. Asserted as long as TBAQ is not empty.
TB_DTD[DBB-1:0]	O	Data out. Request register from TB to DMUT
TB_DTRDY	O	Ready. End of data or command transfer indication. 0 => TB inserts wait state. 1 => TB will finish transfer during this clock cycle.

SMIF interface

BPI_DATA[DBB-1:0]	I	BPI data input
BPI_RD_SFR_N[3:0]	I	BPI read signals
BPI_WR_SFR_N[3:0]	I	BPI write signals
BPI_REQ_N	I	BPI request (asserted with read or write)
TB_BPI_DATA[DBB-1:0]	O	BPI data output
TB_BPI_RDY_N	O	BPI ready (asserted in data cycle of read/write access)

FPI Slave Interface (alternatively to SMIF interface)

TFPI_SEL_N	I	Slave select.
TFPI_A[TFPIAB-1:2]	I	Address bus.
TFPI_D[DBB-1:0]	I	Input Data. Active during data phase of write cycle.
TFPI_WR_N	I	TFPI write to TB
TFPI_RD_N	I	TFPI read TB
TFPI_RDY	I	TFPI ready input
TB_TFPI_RDY	O	Ready. End of transfer indicator: 0 => Master should insert wait states 1 => TB will complete transfer in this cycle
TB_TFPI_RDY_EN	O	RDY output enable
TB_TFPI_D[DBB-1:0]	O	Output Data. Active during data phase of write cycle.

Table 5
Macro Interfaces and Signal Description (cont'd)

Symbol name	I/O	Function
TB_TFPI_D_EN	O	D output enable

Interrupt Controller (IC) Interface

ICTBGNT_N	I	Grant Line
TB_ICREQ_N	O	Request Line
TB_ICD[DBB-1:0]	O	TB interrupt vector data
TB_ICD_EN	O	TB interrupt vector data enable

3.2 Data Flow and Functional Timing

3.2.1 TB Interface to the Protocol Machine Transmit (PMT)

The PMT initiates an address cycle by asserting the read signal PTRD_N. TB captures the channel number from the address bus PTA during this cycle. During the data phase, PMT captures the data as soon as possible. TB asserts TB_PMTRDY during the clock cycle in which it can complete the data transfer.

Two out-of-band signals are required

1.TB_STAT to indicate if transferred word is status instead of pure data. Captured by PMT during TB_PTRDY.

2.TB_EMPTY to show PMT when the TB has no channel data stored. Also captured by PMT during TB_PTRDY (TB_PTD and TB_PTSTAT are not valid).

TB inserts 1 upto 4 waitstates after PTRD_N. Therefore TB can handle a PTRD_N each fifth cycle.

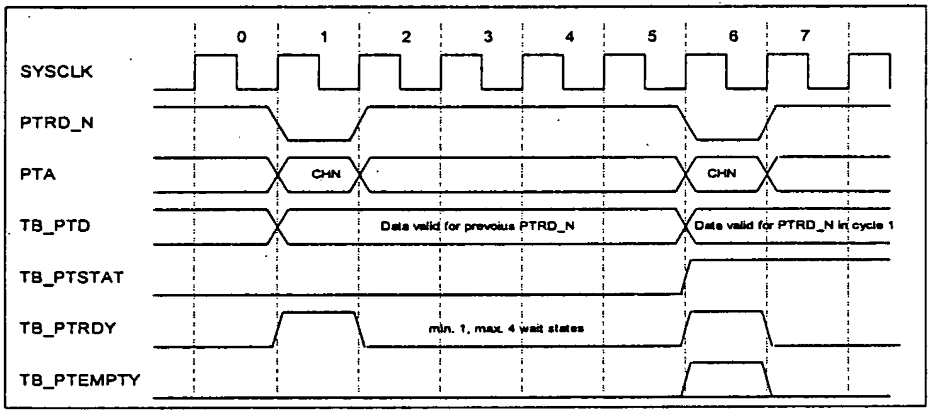

0
1
2
3
4
5
6
7
SYSCLK
PTRD_N
PTA
CHN
CHN
TB_PTD
Data valid for prevoius PTRD_N
Data valid for PTRD_N in cycle 1
TB_PTSTAT
TB_PTRDY
min. 1, max. 4 wait states
TB_PTEMPTY

Figure 5
Read Access from PMT to TB

3.2.2 TB Interface to DMU Controller Receive (DMUT)

The TB interface to the DMUT is based on a bidirectional FPI slave bus. TB initiates a request register access from DMUT by activating TB_REQ_N. The read select signal SEL_N is implicitly active and not physically present. Also OPC is physically not present. DMUT sees TB as a request register at address 0 and a FIFO data port at address 1. Only 1 address bit is defined on the bus.

Two out-of-band signals are required:

1.TB_REQ to indicate that Action Queue FIFO is not empty.

2.DTSTAT to indicate if word in the burst is status instead of pure data. This signal is only valid for current data in DMU transfer.

Data request fromTB: After asserting TB_DTREQ_N DMUT initiates an address cycle (address 0). During the data cycle, TB returns the request register value (channel number and number of requested data) and asserts TB_DTRDY (possible wait states).

After preparing transmit data/status DMUT initiates an address cycle (address 0). During the data cycle DMUT writes channel information and transferlength back to request register. If last word contains frame end or abort indication CI bit is active. DMUT then transfers BL words with overlapped cycles. TB will insert wait states. In best case, a burst cycle can occur with 2 wait states but internally, the PMT and configuration interfaces have higher priority. In the following figures, a burst request from TB to DMUT (**Figure5**) and a data transfer from DMUT to TB is shown. TB may provide several burst requests for different and even the same channel (compare chapter , HOLD condition). DMUT (**Figure 6**) inserts several wait states after writing served burst length and channel number to request register at address 0. An ideal transfer with no collision with a PMT request (2 wait states) an one access with a collision is shown (3 wait states). TB inserts wait states by delaying TB_DTRDY.

The efficiency of the DMU transfers improves with burst length. The peak throughput (not sustainable) can approach 1 word/3 clock with long bursts. However it is ultimately limited by the PMT interface access of the internal buffer RAM. Each word transferred during a TB-DMUT data cycle inserts up to 2 wait states by TB_DTRDY.

Note that DMUT must always read and write the request register for a burst data transfer.

After setting up a new buffer, TB will request data from DMUT by it's own. This request could be used as a 'command acknowledge' indication to generate a command complete interrupt (e.g. done by DMUT). In case of a 'buffer delete' ('transmit off') command, TB discards all stored data and sends a DEL command to DMUT (RTL is not valid). This request could also be used as a 'command acknowledge' indication to generate a command complete interrupt (e.g. done by DMUT).

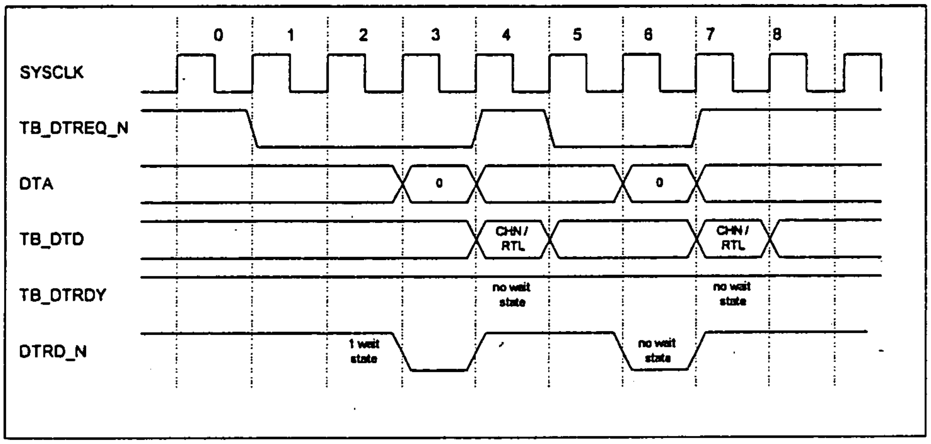


Figure 6
Burst request from TB to DMUT

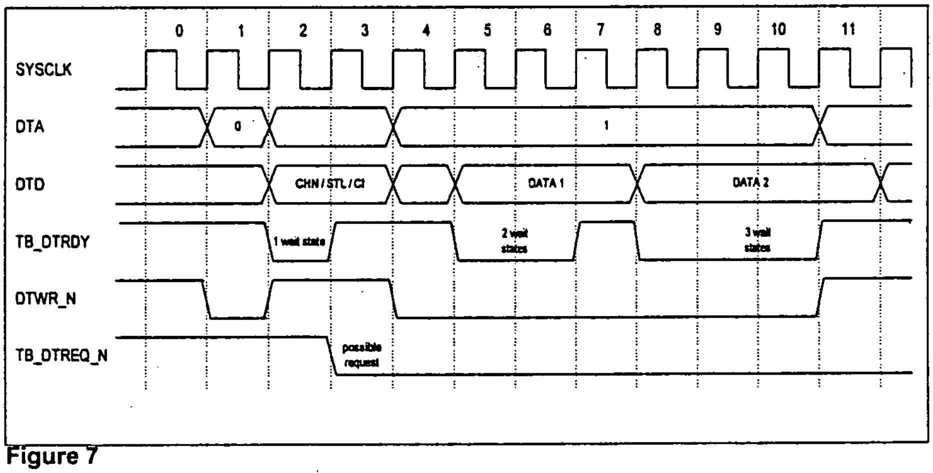


Figure 7
DMUT to TB: 2 data word transfer with wait states

3.2.3 TB Interface to FPI Configuration and Control Bus (FPI Slave) or SMIF interface

The FPI slave interface provides read and write access to all internal data and RAM. It also provides the programming interface for channelwise buffer size (ITBS) and threshold values (BTC, TTC). To change a channel configuration (buffer size and threshold) a configuration command has to be written to command register. According to the programmed values TB controller deletes or sets up a buffer. Only in case of an error condition (not enough free buffer locations for new ITBS) an interrupt is requested to DMUI. Several configuration commands can be written very fast and stored in a configuration fifo if GCSTOP is asserted. In this case, all other interfaces are deactiveated. All configuration commands are executed according to the fifo entries.

FPI Slave interface also provides system test capabilities. Each ram location can be read and written for test purposes. For reading (and writing) rams an autoincrement function is implemented: TYPE in command register (**chapter 4**) specifies the ram and the autoincrement feature interprets all data register read (write) as a read (write) data at next ram address.

The command address is implementation dependent. Figure 7 shows an implementation with command address '0' for channel number, address '1' for channel parameter ITBS, TTC and BTC.

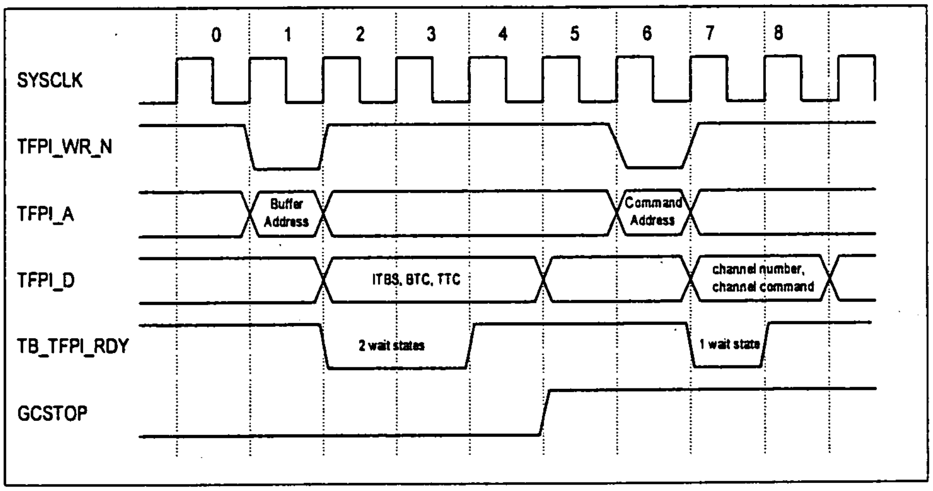


Figure 8
Configuration of a new channel

Note: TFPI interface is an 'add on'. TB macro offers a SMIF register interface with read/ write signals for all readable/writeable register. As via TFPI, a transmit init/off command is executed with a write of the channel specification command register/address, a transmit debug command is excuted with a read of channel specification buffer register after a previous write of the debug command.

3.2.4 Interrupt Controller Interface (IC)

TB activates TB_IC_REQ_N to indicate an interrupt vector on TB_IC_D. Data are valid in cycle after IC_DTGNT

Table 6
Interrupt Vector

Bit	P1	P2			P3			P4	P5
Function	IID		reserved		CDMF	reserved		CHN	

IID: interrupt ID (parameter)

CMDF: command fail interrupt (bit position is parameter)

CHN: channel number

P!: tb_ic_did_lb_c
P2: tb_ic_did_rb_c
P3:`tb_ic_derr_c
P4: tb_ic_dchn_lb_c
P5: tb_icd_dchn_rb_c

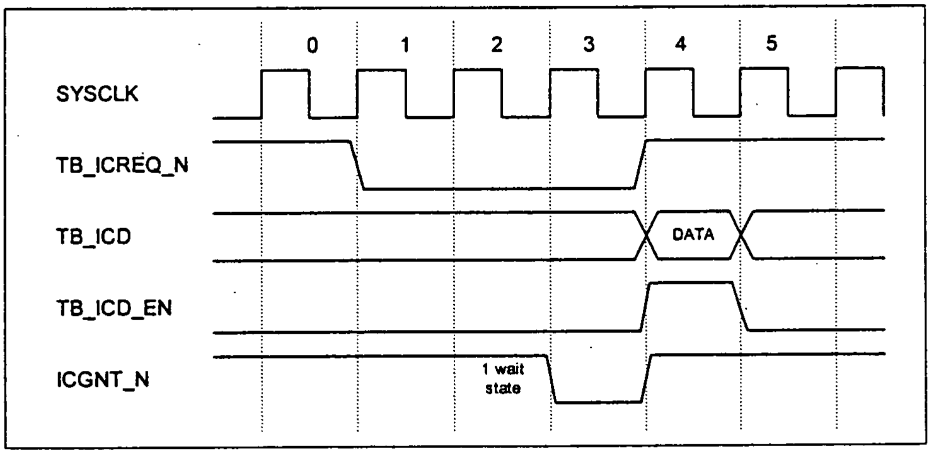


Figure 9
Interrupt Controller Interface

4 Register Description

4.1 Register Overview

Registers have to be provided to configure the channels (FIFO size, threshold value and channel number) and to read and write the internal RAMs (indirect access). and channel parameters.

4.2 Detailed Register Description

Configuration registers can be gathered togehter (TTC, BTC, ITBS, CHN, CFM in one single register) or divided in several registers implementation dependent. Bit positions are also implementation dependent.

Following solution shows 5 register solution; with each write of the channel command register, a new configuration command is written to a configuration FIFO in TB.

Note 1: A new configuration is started with write of channel command register. To programm all channels in the same manner, only for one channel the TTC, BTC, ... registers have to be written!

Note 2: After a successfull 'buffer create' command (i.e. buffer exists) first a 'buffer delete' command' has to be given, before the next 'buffer create' can be programmed. For details see chapter 2.

Note 3: 'P' indicates that this bit position/ bus width is a parameter.

4.2.1 Channel Command Register

Access : write

Address : *tb_vcs_cmd_adn_c*

Reset Value : 00000000_H

	P1	P2	P3		P4 P5
Reserved	init	off	debug	Reserved	CHN

CHN (number of bits and position are parameters):

channel number

CMD (number of bits and position are parameters):

channel command: TB provides 3 commands

- buffer create ('transmit init'),
- buffer delete ('transmit off'),
- buffer parameter debug ('transmit debug') command.

P1: tb_vcs_init_c

P2: tb_vcs_off_c
P3: tb_vcs_debug_c
P4: tb_vcs_chn_lb_c
P5: tb_vcs_chn_rb_c

4.2.2 Buffer Parameter Register

Access	: write/read
Address	: *tb_cvs_bufs_adn_c*
Reset Value	: 00000000_H

	P1 P2		P3 P4		P5 P P6
reserved	TTC	reserved	BTC	reserved	ITBS

BTC (number of bits and position are parameters):

Burst Threshold Code.

0: burst threshold = 1 word

others: burst threshold = $2^{(BTC+1)}$ words (as an example, coding is also a parameter)

TTC (number of bits and position are parameters):

Transmit Threshold Code.

0: burst threshold = 1 word

others: burst threshold = $2^{(TTC+1)}$ words (as an example, coding is also a parameter

ITBS (number of bits and position are parameters):

Individual Transmit channel Buffer Size = ITBS words

P1: tb_tb_vcs_ttc_lb_c
P2: tb_tb_vcs_ttc_rb_c
P3: tb_tb_vcs_btc_lb_c
P4: tb_tb_vcs_btc_rb_c
P5: tb_tb_vcs_itbs_lb_c
P6: tb_tb_vcs_itbs_rb_c

4.2.3 Indirect Access Register: Address

All rams and some register can be read/written by writing 'indirect access register address'. If autoincrement function is selected, with each read/ write access to 'indirect access register data' the next ram address is read/written. Only a startaddress has to be defined. Autoincrement may be stopped (no read/write) or interrupted (by new TYPE definition).

Access	: read/write

Address : *tb_tac_adn_c*
Reset Value : 00000000_H

P1 P2 P3 P4 P5

MID	reserved	AINC	reserved	COMMAND

P6 P7

Channel#/Address

MID:

Macro Idenitfication Number; access defined by TYPE etc. is only performed if MID matches the (implementation specific) ID.

AINC:

Automatic Incrementation

AINC=1 : The address will be automatically incremented with each read or write access

AINC=0 : The address won't be incremented (default value).

P1: tb_vg_mid_lb_c
P2: tb_vg_mid_rb_c
P3: tb_vg_ai_c
P4: tb_vg_cmd_lb_c
P5: tb_vg_cmd_rb_c
P6: tb_vg_adr_lb_c
P7: tb_vg_adr_rb_c

COMMAND:

tb_vg_cmd_fpc_c: read free pool counter
tb_vg_cmd_fpp_c: read free pool pointer
tb_vg_cmd_gfpc_c: read global free pool counter
tb_vg_cmd_aqctr_c: read action queue counter
tb_vg_cmd_cqctr_c: read configuration queue counter
tb_vg_cmd_pmt_on_c: switch pmt interface on
tb_vg_cmd_pmt_off_c: switch pmt interface off
tb_vg_cmd_pmt_c: switch to pmt interface mode
tb_vg_cmd_dmut_on_c: switch dmut interface on
tb_vg_cmd_dmut_off_c: switch dmut interface off
tb_vg_cmd_dmut_req_c: select dmut request register
tb_vg_cmd_dmut_dat_c: select dmut data register
tb_vg_cmd_aq_c: read/write action queue ram

tb_vg_cmd_cq_c: read/write configuration queue ram
tb_vg_cmd_pt0_c: read/write parameter table ram
tb_vg_cmd_pt1_c: read/write parameter table ram
tb_vg_cmd_pt2_c: read/write parameter table ram
tb_vg_cmd_pt3_c: read/write parameter table ram
tb_vg_cmd_ll_c: read/write link list ram
tb_vg_cmd_db_c: read/write data buffer ram
tb_vg_cmd_stat_c: read/write status bit ram

Channel#/Address:

Channel number or Address field

4.2.4 Indirect Access Register: Data

Access	: read/write
Address	: *tb_td_adn_c*
Reset Value	: 00000000$_H$

31	16
Test Data	

15	0
Test Data	

Test Data:

Data from/for different Rams / Free Pool Counter (number of unused buffer locations).

6 Appendix: M256F tb_shell

For M256F application a special naming convention is used. Addtional 'daisy chain' signals are added.

6.1 Register Description

6.1.1 Register Overview

Table 7PCI Slave Register Set (Direct Addressing)

Register Name		Read/ Write	Offset to PCI BAR1
Virtual Global Registers			
	TAC	R/W	058_H
	TD	R/W	$05C_H$
Macro Specific Registers			
Virtual Channel Specification Registers			
	CSPEC_CMD	W	000_H
	CSPEC_BUFFER	R/W	020_H

6.1.2 Detailed Register Description

6.1.2.1 (Virtual) Channel Specification Command (CSPEC_CMD)

Access	: write/read (only CHAN readable)
Address	: 000_H
Reset Value	: 00000000_H

31	16
CMD_XMIT(7:0)	CMD_REC(&:0)

15 8	7 0
00	CHAN(7:0)

Note: The Virtual Channel Spec (VCS) Command Register has to be programmed after all other required VCS registers, in order to initiate the programming of all macros. In case of a debug command, first the command register has to be written, then a broadcast read of the virtual channelspec is possible by read of all VCS data registers. Only the macro which has implemented the corresponding bit has to drive the register bit, otherwise drives '0' to provide broadcast read feature.

CHAN(7:0): selected channel number to be programmed

CMD_XMIT(7:0): for details refer to table "Command Description"

Note: Transmit Init for a channel must be programmed only after reset or after a Transmit Off command, i.e. two Transmit Init commands for the same channel are not allowed

Command Table Transmit:

31	30	29	28	27	26	25	24	
Rsvd	Rsvd	Idle	Debug	HOLD RESET	ABORT	OFF	INIT	function
0	0	0	0	0	0	0	1	transmit init
0	0	0	0	0	0	1	0	transmit off
0	0	0	1	0	0	0	0	transmit debug
0	0	0	0	0	0	0	0	transmit nop

6.1.2.2 (Virtual) Channel Specification Buffer (CSPEC_BUFFER)

Access	: read/write
Address	: 020_H
Reset Value	: 00000000_H

31 29	28 16
~~TQUEUE(2:0)~~	ITBS(12:0)

15 12		10 8		6 4		2 0
TTC(3:0)	0	TBTC(2:0)	0	~~RQUEUE(2:0)~~	0	~~RBTC(2:0)~~

TBTC: transmit.burst threshold code

ITBS: individual transmit buffer size

TTC: Transmit Threshold Code

(for coding see chapter 6.4)

6.1.3 Test Command Register (TAC)

Access	: read/write
Offset Address	: $_H$
Reset Value	: 00000000$_H$

31 28				24	23 16
MID	0	0	0	AI	CMD

15			12 0
0	0	0	ADDRESS

MID:	Macro ID Code
AI:	Auto Increment Function
Address:	internal address
CMD:	Command (select of ram and register)

CMD:

23	22	21	20	19	18	17	16	
								function
0	0	0	0	0	0	0	0	tb_vg_cmd_fpc_c
0	0	0	0	0	0	0	1	tb_vg_cmd_fpp_c
0	0	0	0	0	0	1	0	tb_vg_cmd_gfpc_c
0	0	0	0	0	0	1	1	tb_vg_cmd_aqctr_c
0	0	0	0	0	1	0	0	tb_vg_cmd_cqctr_c
1	0	1	0	1	1	1	0	tb_vg_cmd_pmt_on_c
1	0	1	0	1	0	1	0	tb_vg_cmd_pmt_off_c
1	0	1	0	0	0	0	0	tb_vg_cmd_pmt_c

23	22	21	20	19	18	17	16	
1	0	1	1	1	1	1	0	tb_vg_cmd_dmut_on_c
1	0	1	1	1	0	1	0	tb_vg_cmd_dmut_off_c
1	0	1	1	0	0	0	1	tb_vg_cmd_dmut_req_c
1	0	1	1	0	0	0	0	tb_vg_cmd_dmut_dat_c
1	1	1	1	0	0	0	0	tb_vg_cmd_aq_c
1	1	1	1	0	0	0	1	tb_vg_cmd_cq_c
1	1	1	1	0	0	1	0	tb_vg_cmd_pt0_c
1	1	1	1	0	0	1	1	tb_vg_cmd_pt1_off_c
1	1	1	1	0	1	0	0	tb_vg_cmd_pt2_c
1	1	1	1	1	0	0	0	tb_vg_cmd_pt3_c
1	1	1	1	0	1	0	1	tb_vg_cmd_ll_c
1	1	1	1	0	1	1	0	tb_vg_cmd_db_c
1	1	1	1	0	1	1	1	tb_vg_cmd_stat_c

Macro ID Code:

31	30	29	28	macro
0	0	1	1	TB (tb_vg_mid_tb_c)

General

- the test access provides read/write access of important internal rams and registers
- test registers are virtuals global registers (SEL signal: pb_vg_tfpi_sel_n / implemented as a daisy chain).
- the single macros are selected by the MID code of the test command register
- CMD specifies/selects one of the macro rams/registers
- the address field is used to access a ram address
- AI: autoincrement : address given in the address field is incremented automatically for each access
- All macros which are not selected by MID drive data output "00000000" and <macro>_TPFI_RDY = '1'. Driving "00000000" would mean not disable the enable line for data out, but to set the output data to "00000000".

Test write access:

1. Write TAC
2. Write TAD

Test read access:

1. Write TAC
2. Read TAD

Typically the macro selected via MID delays the RDY signal until the selected ram/ register has been read and the data can be provided at the TFPI interface. No prefetch of testdata is required.

Note: CMD/Address have to be defined for each macro; there is no read/write selection in the CMD field; rd/wr's are handled with the TFPI read & write signals

6.1.4 Test Data Register (TD)

Access	: read/write
Address	: $5C_H$
Reset Value	: 00000000_H

31	
TEST DATA	

15	0
TEST DATA	

TEST DATA:	ram/register test data (read or write)

6.2 DMUT interface

Table 8
Request Register for data request from TB to DMUT

Bit	31			28	16			7	0
Function	DEL	Reserved		RTL (requested transfer length)		Reserved		CHN (Channel number)	

Table 9
Request Register for data transfer acknowledge from DMUT to TB

Bit	31			28	16			7	0
Function	CI	Reserved		STL (served transfer length)		Reserved		CHN (Channel number)	

6.3 Interrupt Controller Interface (IC)

Table 10
Interrupt Vector

Bit	31	28			17			7	0
Function	1001		0		CDMF	0		CHN	

6.4 Ram sizes

6.4.1 TB_PTR (TB_PTR1)

Parameter Table: 256 x 89

Table 11: All channel parameters:

ITBS	BTC	TTC	FIRST	AQE	WAIT	CIPV	CIP	FCTR	ECTR	WP	RP
(13)	(3)	(4)	(1)	(1)	(1)	(1)	(13)	(13)	(13)	(13)	(13)

6.4.2 TB_DBR and TB_LLR

Data Buffer: 8K x 33

Link List: 8K x 13

6.4.3 TB_AQR

Action Queue: 256 x 8

6.4.4 TB_CFGR (TB_RTR2)

Configuration Queue: 256 x 28

Table 12: All locations:

ITBS	BTC	TTC	CHN
(13)	(3)	(4)	(8)

6.5 Shell Interface and Signal Description

M256F Symbol name	I/O	Macro Symbol Name	Function

Clock and Reset

SYSCLK	I	SYSCLK	Internal system clock (33 MHz)
HW_RESET_N	I	(RESET_N)	General hardware reset of TB. All registers and RAM reset or initialization.
SW_RESET_N	I	-	General sofrware reset of TB. All registers and RAM reset or initialization.
SCANMODE	I	-	Scanmode
GC_STOP	I	GCSTOP	Stop all non configuration process in TB (fast programm mode)
TB_IIP	O	TB_IIP	Initialization of TB rams in progress

Protocol Machine Receive (PMT) Interface

PT_TB_RD_N	I	PTRD	PMT read. Only single word write transfer is supported.
PT_TB_A [CNB-1:0]	I	PTA	Address bus. Specifies channel number for transfer.
TB_PT_D [DBB-1:0]	O	TB_PTD	TB Data/Status word being transferred to PMT.
TB_PT_RDY	O	TB_PTRDY	Ready. End of data transfer indication. 0 => TB inserts wait state. 1 => TB will finish transfer during next clock cycle.
TB_PT_STAT	O	TB_PTSTAT	Mode of data word to be transferred. 0 => protocol data 1 => status + protocol data

M256F Symbol name	I/O	Macro Symbol Name	Function
TB_PT_EMPTY	O	TB_PTEMPTY	Asserted for one cycle while PMT read if TB has no data stored for the channel specified by the address bus.

DMU Transmit (DMUT) Interface

DT_TB_RD_N	I	DTRD	Read. DMUT Read Control (for request register)
DT_TB_WR_N	I	DTWR	Write. DMUT Write Control (for request register or data register)
DT_TB_A	I	DTA	Address bus (1 bit) 0 => Request register. 1 => Data register.
DT_TB_D[DBB-1:0]	I	DTD	Data in. Input command from DMUT (channel number e. g.) or data/ status input
DT_TB_STAT	I	DTSTAT	Status indication from DMUT
TB_DT_REQ_N	O	TB_DTREQ	Service request from TB to DMUT controller. Asserted as long as TBAQ is not empty.
TB_DT_D[DBB-1:0]	O	TB_DTD	Data out. Request register from TB to DMUT controller
TB_DT_RDY	O	TB_DTRDY	Ready. End of data or command transfer indication. 0 => TB inserts wait state. 1 => TB will finish transfer during this clock cycle.

Interrupt Controller (IC) Interface

M256F Symbol name	I/O	Macro Symbol Name	Function
TB_IC_REQ_N	O	TB_ICREQ	Request Line
IC_TB_GNT_N	I	ICTBGNT	Grant Line
TB_IC_D[31:0]	O	TB_ICD	TB interrupt vector data
IC_D	I	-	daisy chain data input

FPI Slave Interface

PB_TFPI_RD_N	I	TFPI_RD_N	TFPI read TB
PB_TFPI_WR_N	I	TFPI_WR_N	TFPI write to TB
PB_TFPI_A[8:2]	I	TFPI_A[TFPIAB-1:2]	Address bus.
PB_TFPI_D[31:0]	I	TFPI_D	Input Data. Active during data phase of read cycle.
PB_TFPI_RDY	I	TFPI_RDY	TFPI ready input
PB_VC_TFPI_SEL_N	I	TFPI_VC_SEL_N	Slave select.
PB_VG_TFPI_SEL_N	I	TFPI_VG_SEL_N	Slave select.
TFPI_D[31:0]	O	-	Daisy chain input.
TB_TFPI_RDY	O	TB_TFPI_RDY	End of transfer indicator: 0 => Master should insert wait states 1 => TB will complete transfer in this cycle
TB_TFPI_D[31:0]	O	TB_TFPI_D	Output Data. Active during data phase of write cycle.

6.6 Dimension of M256F application:

Name	DescriptionO	M256F value
tb_chn_nw_c	number of supported channels	256
tb_chn_nb_c	channel bus bits	8
tb_db_nw_c	depth of data buffer	8192
tb_dba_nb_c	data buffer address bus bits	13
tb_btc_nb_c	number of BTC bits	3

tb_ttc_nb_c	number of TTC bits	4
tb_btc2btl	coding of TTC values	0: 1 1: 4 2: 8 3: 16 4: 32 5: 64 6: 128 7: 256
tb_ttc2ttl	coding of TTC values	0: 1 1: 4 2: 8 3: 12 4: 16 5: 24 6: 32 7: 40 8: 48 9: 64 A: 96 B: 128 C: 192 D: 256 E: 384 F: 512
tb_data_nb_c	data bus width	32
tb_test_pt*_**_c (*: 1,2,3 and **: rb,lb)	selected slices for testmode access of parameter table ram	1: lb=63/rb=32 2: lb=88/rb=64 3: lb=88/rb=64 (3 not used)